IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of		:	
NEIL W. TAYLOR		:	
Serial No.:	09/862,828	:	Group Art Unit: 2135
Filed:	May 22, 2001	:	Examiner: Son, Linh L D
For:	METHODS FOR DETECTING EXECUTABLE CODE WHICH HAS BEEN ALTERED		

PRE-APPEAL BRIEF REQUEST FOR REVIEW FILED CONCURRENTLY WITH NOTICE OF APPEAL AND NOTICE OF APPEAL FEE

Mail Stop After Final (AF)
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Responsive to the Final Office Action dated April 6, 2006, the Applicant hereby requests a pre-appeal conference of a panel of Examiners.[1]

By this response, claims 1-4 and 7-22 are pending. Substantively, the Examiner rejects all claims as obvious over U.S. Patent No. 5,493,649 to Slivka in view of U.S. Patent No. 5,944,821 to Angelo. Yet, the pending claims have many features and aspects not found in the combination of references. Also, the Examiner has failed to show proper motivation to combine the references and, among other things, wrongly uses the invention as a roadmap to reconstruct the teachings of the past. In organization, it appears the Examiner contends Slivka teaches all elements of the claims with the exception that Angelo teaches "one of randomly and substantially each time the executable code is launched for use, calculating a plurality of subsequent scores on the executable code." *See, 4-6-06 Final Office Action, p. 3.* In other words, the Examiner relies on Slivka for anticipating all elements of the claims except for the aspect of calculating plural subsequent scores randomly or substantially each time the code is launched for use, to which the Examiner cites Angelo. As is apparent below, this analysis is deficient.

---

[1] Further, this request is in view of the twice rejection of the pending claims and pursuant to the Pilot Program established in the Official Gazette, dated July 12, 2005, and since extended.

For instance, Slivka concerns itself with a comparison of one or more checksums relative to either code or data. As in previous papers, only the checksums of the "code section" of Slivka are relevant to the instant invention.[2] With this in mind, Slivka calculates a checksum for the "code section" at step 402. At step 410, a "new code checksum" is calculated. In the event the first and new "code" checksums are not equivalent, corruption is indicated at step 428. Then, "processing ends, before the code is executed." *Col. 3, ll. 57-58.* Yet, this checksum comparison is wholly a one-to-one comparison of a first checksum to a second or new checksum. It is unequivocally never the calculation of "an initial score" and a calculation of subsequent "plural scores," much less subsequent exclusive comparison of "each" of the subsequent plural scores to the initial score "and to no other scores," as representatively claimed in Applicant's independent claim 1. In other words, Slivka's one-to-one comparison of a first checksum to a second checksum does not have multiple instances of subsequent checksums nor does it have exclusive comparisons between the multiple instances to the first checksum and to no others. As a matter of law, this cannot anticipate.

To the extent Slivka, indeed, calculates third or more "code section" checksums (indicated in one instance by off-page connector "C" being introduced back into the flow diagram of Figure 4a before step 406), this third or more code section checksum, as well as the "new" or second code section checksum at step 410, is analyzed at step 408 to determine "whether a periodic interval has elapsed." *Col. 3, ll. 44-45.* According to Slivka, a periodic interval "can be based on timing, count of operations requested, or other suitable events." *Col. 3, ll. 45-46.* Purportedly, "modifications to the code section are **very rare**." *Emphasis added, col. 3, l. 40.* In turn, it is "prefer[red]" that code section checksums not be compared or "checked upon receipt of every operation request." *Col. 3, ll. 42-43.* In other words, Slivka believes the rarity of changes to code sections of programs, for example, enables a delay to be introduced in the code section checksum comparison routine and it

---

[2] *See, Applicant's Amendment and Remarks response dated January 16, 2006.* Alternatively, the Applicant presents prior arguments relative to Slivka's "data section" checksum teachings. In this regard, Slivka calculates a first data section checksum at step 404 and compares it at step 418 to a "newly" calculated data section checksum from step 416. Yet, this is another instance of a one-to-one comparison of a first checksum to a second or new checksum and fails to anticipate claims requiring multiple second scores exclusively being compared to an initial score. To the extent Slivka makes other, incremental checksums at step 426, for instance, this relates to writing operations. Subsequent checksum comparisons are then made back to the immediately one prior checksum, not the original or first data section checksum. In turn, this data section checksum feature cannot anticipate claims precisely requiring "exclusive" comparisons of plural subsequent scores back to an "initial"score and "to no other scores." *See, e.g., Applicant's independent claim 1.*

is a best practice to prevent code section checksum comparisons from occurring all the time. Otherwise, processing would bog down. Expressly, the "periodic interval" routine is the mechanism for introducing delay and such is regular or periodic during code section checksum comparisons.

In great contrast, all claims of the instant invention require calculating plural subsequent scores nearly "each time" the "executable code is launched for use" after the initial loading of the executable code. Alternatively, all claims of the instant invention require calculating plural subsequent scores "randomly."[3] In this manner, the Applicant's invention contemplates thwarting hacker attempts on executable code by either being tedious or unpredictable. As indicated throughout the Applicant's Background of the Invention section, it is intensely appreciated that hackers have become extremely adept over the years and now possess exceptional talent for inserting malicious code and/or viruses into executable code. Whereas, heretofore, they were essentially unconcerned or novices at it. To wit, Slivka's teaching characterizes alterations or modifications to code as "very rare" and thus only worthy of receiving "periodic interval" checksum comparisons. The Applicant's claims, however, are not so precluded. Rather, the Applicant's invention far exceeds Slivka's contemplated hacker prevention and all claims are submitted as allowable.

Additionally, the Applicant's claims 11 and 12 precisely recite that calculations of subsequent scores are doubly performed both "randomly and substantially each time the executable code is launched for use." Nowhere does Slivka contemplate or even appreciate this sophisticated doubly secure methodology. Claim 12 recites another labyrinth of protection by requiring calculation of plural subsequent scores with "predetermined time intervals" layered upon the doubly secure calculations of both "randomly and substantially each time the executable code is launched for use." Nowhere does Slivka's teaching approach this level of sophistication.

Angelo, on the other hand, does not supply the missing teaching rendering the claims obvious. Rather, Angelo concerns itself with calculating hash values of programs prior to or before the program being loaded into memory and executed. Angelo never discloses scores or calculations

---

[3] Support for these limitations are found throughout the specification. Especially, *p. 15, ll. 9-18.*

of scores associated with executable code at a time when the code is "initially or shortly thereafter loaded into an operating system."[4]

In all embodiments, Angelo's hash value comparisons occur at a time prior to the executable code being installed in an operating system. Representatively, Angelo's Figure 3 shows generating a SMI (step 300); entering the SMM via the CPU (step 302); generating a hash value (step 304); validating the stored hash table (step 306); checking entries in the hash table and verifying same (steps 308-316). Thereafter, it loads the program into memory and executes it (step 318). From the usage of antecedent language in the steps, it should be appreciated that step 304 requires the generation of a secure hash value for a "program to be executed." Then, at step 318, it is this same "program to be executed" that is subsequently "loaded into memory and executed." At *col. 9, ll. 6-8*, relative to Figure 2, Angelo reiterates this concept by stating the hash algorithm "securely register[s] and verif[ies] the integrity of software applications prior to execution." *Underlining added*. In other words, Angelo teaches hash value generation and comparison at a time prior to or before loading executable code (having its hash value generated) into an operating system. This, the Applicant respectfully submits, cannot then as a matter of law be used in rejecting claims requiring the contrary. *See also, Applicant's prior paper entitled AF-Request for Reconsideration, dated June 14, 2005.*

Alternatively, Angelo disparages prior art solutions that attempt to solve problems of code modification detection via software products and/or in other than protected memory areas, such as SMM memory. For example,

> [i]n some earlier systems, a secure hash value is calculated and stored for newly installed software. Thereafter, when the computer is turned on again, the stored hash value is compared to a newly calculated value. If a discrepancy is found, the user is alerted. A main disadvantage with this method is that the integrity assessment codes must be stored on the hard disk, thus making the codes themselves susceptible to attack by malicious code. Reverse-engineering a modification detection code, while difficult, is not a mathematically intractable problem. Thus, software-only protective products can

---

[4] This claim language is expressly found in claim 1. Of course, the other claims have related language but their slight variations change the overall scope of the claims. In prior papers, the Applicant also touts aspects of the instant invention as including this timing of when the code is loaded.

> offer only limited insurance against the attack of malicious code, due mainly to architectural weakness present in most computer systems. A potential solution is to embed the modification detection code in a permanent read-only memory device, but this can make system reconfiguration quite difficult. *Col. 2, l. 60 - col. 3, l. 8.*

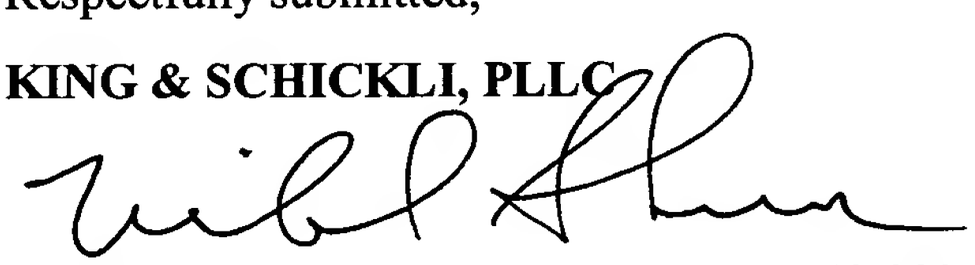
Thus, Angelo teaches away from the instant invention, especially in instances, such as claims 21 and 22, requiring computer readable medium. Angelo teaches away from Slivka because Slivka clearly operates in memory 202 other than protected memory. The combination thusly fails.

Still alternatively, the Applicant has already successfully argued past the teachings of Angelo. Now, with the claims even more precisely defined and facing anticipation rejections from Slivka, the Examiner returns to Angelo to piecemeal together an obviousness rejection. He also selectively culls bits and pieces of the references, without motivation to do so, to fit the claims as he seemingly, whimsically sees fit. His rejections, however, are clearly discouraged under the law.[5]

The Applicant submits all claims are in a condition for allowance and requests a decision indicating same. ***To the extent any fees are due beyond those authorized in the accompanying fee transmittal sheet for the Notice of Appeal, the undersigned authorizes their deduction from Deposit Account No. 11-0978.***

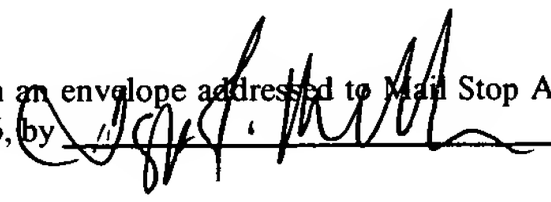
Respectfully submitted,

KING & SCHICKLI, PLLC



Michael T. Sanderson, Registration No. 43,082
247 North Broadway
Lexington, Kentucky 40507
(859) 252-0889

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being sent by First Class Mail in an envelope addressed to Mail Stop After Final (AF), Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 8, 2006, by  .

[5]As is well established, "virtually all [inventions] are combinations of old elements." *Ruiz v. A.B. Chance Co.*, 69 USPQ2d 1686, 1690 (Fed. Cir. 2004). Also, an obvious determination under 35 U.S.C. 103(a) requires an "as a whole" analysis of the prior art to otherwise prevent an impermissible "evaluation of the invention part by part." *Id.* For otherwise, "an obviousness assessment might break an invention into its component parts (A+B+C), then find a prior art reference containing A, another containing B, and another containing C, and on that basis alone declare the invention obvious." *Id.* In turn, "this form of hindsight reasoning, using the invention as a roadmap to find its prior components, would discount the value of combining various existing features or principles in a new way to achieve a new result - often the very definition of invention." *Id.*

AF
JFW

Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional)
1363-004

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]

on May 8, 2006

Signature

Typed or printed name Tanya J. Mirilovich

Application Number	Filed
09/862,828	05/22/01

First Named Inventor
Neil W. Taylor

Art Unit	Examiner
2135	Linh L D Son

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☑ attorney or agent of record. Registration number 43,082

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

Signature

Michael T. Sanderson
Typed or printed name

859/ 252-0889
Telephone number

5-8-06
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☑ *Total of 3 forms are submitted.